G01: ปัญหาการเลือกกิจกรรม (Activity Selection Problem)

ให้นักเรียนเขียนโปรแกรมเพื่อหาจำนวนกิจกรรม โดยกำหนดให้มีห้องหนึ่งห้อง มีกิจกรรมที่จะต้องใช้ห้อง หลายกิจกรรม แต่ละกิจกรรมจะมีเวลาเริ่มต้น และเวลาสิ้นสุด ต้องการเลือกกิจกรรมให้ได้มากที่สุด โดยแต่ละ กิจกรรมที่ถูกเลือก ต้องมีเวลาไม่ทับซ้อนกัน ยกเว้นเวลาสิ้นสุดและเวลาเริ่มต้นของกิจกรรมต่อไป สามารถเป็น เวลาเดียวกันได้

ข้อมูลนำเข้า บรรทัดที่ 1 n แทนจำนวนกิจกรรมที่มีทั้งหมด

บรรทัดที่ 2 ถึงบรรทัดที่ n+1 ประกอบด้วยจำนวนเต็ม 2 จำนวน (s, f)

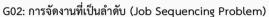
จำนวนแรก s_i แทนเวลาเริ่มต้นของกิจกรรมที่ i

จำนวนแรก f_i แทนเวลาสิ้นสุดของกิจกรรมที่ i

ข้อมูลนำออก บรรทัดที่ 1 N แสดงจำนวนกิจกรรมที่มากที่สุดที่เลือกได้

บรรทัดที่ 2 ถึงบรรทัดที่ N+1 แสดงรายการกิจกรรมที่ถูกเลือก

Input	Output
11	4
0 6	1 4
1 4	5 7
2 13	8 11
3 5	12 14
3 8	
5 7	
5 9	
6 10	
8 11	
8 12	
12 14	



กำหนดอาร์เรย์ของงานมาให้ โดยที่ทุก ๆ งานจะมี deadline และกำไร (profit) จากงานนั้น ๆ ถ้างานนั้นทำ เสร็จ ต้องการจัดงานให้เป็นไปตามกำหนดเวลา โดยได้กำไรที่มากที่สุด

ตัวอย่าง

Input: Four Jobs with following deadlines and profits			Inp	out: Fiv	e Jobs w	th foll	owing deadlines and profits	
JobID	Dead	lline	Profit		JobID	Deadline	e Pr	rofit
a	4	20			a	2	100	
b	1	10			b	1	19	
С	1	40			С	2	27	
d	1	30).		d	1	25	
Output	: Follov	ving is n	naximum profit sequence of jobs		е	3	15	
c, a				Οι	ıtput: Fo	ollowing i	s maxi	mum profit sequence of jobs
				a,	c, e			

ให้นักเรียนเขียนโปรแกรมเพื่อแก้ปัญหาการจัดงานที่เป็นลำดับ (Job Sequencing Problem)

Input	Output
4	са
a 4 20	
b 1 10	
c 1 40	
d 1 30	
5	асе
a 2 100	
b 1 19	
c 2 27	
d 1 25	
e 3 15	

G03: การทอนเงินให้จำนวนเหรียญน้อยที่สุด (Minimum number of coins)

ให้นักเรียนเขียนโปรแกรมเพื่อแก้ปัญหาการทอนเงิน โดยแคชเชียร์มีเหรียญ 1 บาท, 2 บาท, 5 บาท, และ 10 บาท จำนวนไม่จำกัด (ไม่มีธนบัตร) ต้องการทอนเงิน W บาท โดยใช้จำนวนเหรียญในการทอนน้อยที่สุด ต้องทอน อย่างไร

ข้อมูลนำเข้า w แทนจำนวนเงินที่ต้องทอน

ข้อมูลนำออก แสดงจำนวนเหรียญแต่ละประเภทตามลำดับ คือ เหรียญ 10 บาท, 5 บาท, 2 บาท,

และ 1 บาท

Input	Output
28	2
	1
	1
	1
49	4
	1
	2
	0



ให้นักเรียนเขียนโปรแกรมเพื่อแก้ปัญหา fractional knapsack

ข้อมูลนำเข้า บรรทัดที่ 1 W แทนน้ำหนักที่ถุงจะรับได้มากที่สุด

N แทนจำนวนกอง

บรรทัดที่ 2 ถึงบรรทัดที่ n+1 ประกอบด้วยจำนวนเต็ม 2 จำนวน (v. w)

จำนวนแรก v_i แทนมูลค่าของกองที่ i

จำนวนแรก w_i แทนน้ำหนักของกองที่ i

ข้อมูลนำออก แสดงมูลค่ามากที่สุดของของในถุง

		Input	t	Output
	100 2			100.00
	200 1000			
	100 100			
3	(4)B			180.00
	100 (2)	50	1	
	10 2	40	1	
	120 3	5	1	

$$\left(\frac{x^3}{3} + x - \frac{x^2}{2}\right) \Big|_{1}^{2}$$

นมางถึง
$$\left(\frac{2^3}{3} + \times - \frac{2^2}{2}\right) \left(\frac{1^3}{3} + 1 - \frac{1^2}{2}\right)$$

โดนำ 2 มาเทษ
ภากนึ้ง ค่างถ้างแบ่ง 1

G05: ตารางเรียนออนไลน์ (Course Schedule)

งานที่ต้องทำคือ จัดตารางเรียนออนไลน์ โดยมีคอร์สเรียนออนไลน์ทั้งหมด n คอร์ส (1 ถึง n) แต่ละคอร์สมี ระยะเวลาสอน t และจะปิดคอร์สในวันที่ dth คอร์สดังกล่าวใช้วันสอนต่อเนื่องตามระยะเวลาที่กำหนด สำหรับ t วัน การจัดคอร์สลงตารางเรียน คอร์สแรกที่เลือกเริ่มต้นที่วันที่ 1 และในช่วงเวลาเดียวกับจะไม่มีการจัดคอร์ส ซ้ำซ้อนในเวลาเดียวกัน ให้นักเรียนเขียนโปรแกรมเพื่อหาจำนวนคอร์สที่มากที่สุดที่สามารถจัดลงตารางตาม เงื่อนไขที่กำหนดได้

ข้อมลนำเข้า บรรทัดที่ 1 จำนวนเต็มสองจำนวน n แทนจำนวนคอร์ส โดยที่ $2 \le n \le 100$

> บรรทัดที่ 2 ถึงบรรทัดที่ n+1 จำนวนเต็ม $\,t\,$ และ $\,d\,$ แทนจำนวนวัน (ระยะเวลาที่ใช้ของ คอร์สที่ i) และวันที่ ที่คอร์สต้องสิ้นสดก่อน

โดยที่ $1 \leq t \leq 1000$ และ $1 \leq d \leq 10000$

จำนวนคอร์สที่มากที่สดที่สามารถจัดลงตารางตามเงื่อนไขที่กำหนดได้ ข้อมูลนำออก ตัวอย่าง

Input	Output
4	3
100 200	
200 1300	
1000 1250	
2000 3200	
3	2
2 6	
3 3	
4 4	
3	1
5 5	
4 6	
2 6	

<u>ตัวอย่างที่ 1</u> คอร์สที่ถูกเลือกคือ [100 200] อันดับแรก ถัดมาคือ [1000 1250] โดยเริ่มทำงานในวันที่ 101 เนื่องจากคอร์สแรกที่เลือกใช้เวลา 100 วัน และสิ้นสุดในวันที่ 1001 คอร์สที่ 3 ที่เลือกคือ [200 1300] สิ้นสุดใน วันที่ 1300 ส่วนคอร์ส [2000 3200] ไม่ถูกเลือกเนื่องจากใช้เวลา 2000 วัน ทำให้สิ้นสุดวันที่ 3300 เกินวันสิ้นสุด ที่กำหนด คือ วันที่ 3200 ดังนั้นสามารถเลือกคอร์สจัดลงตารางได้มากที่สุด เท่ากับ 3

<u>ตัวอย่างที่ 2</u> เลือกได้มากสุด 2 คอร์ส คือแบบที่หนึ่ง [3 3] และ [2 6] หรือแบบที่สอง [4 4] และ [2 6] ตัวอย่างที่ 3 เลือกได้มากสุด 1 คอร์ส

G06: Rooms

เจ้าของโรงแรมแห่งหนึ่งต้องการคำนวณหาจำนวนห้องพักที่น้อยที่สุดที่ต้องสร้าง โดยเขามีรายการการเข้าพักของ ลูกค้าประกอบด้วย วันที่มาถึง และจำนวนวันที่ลูกค้าต้องการพัก ทั้งนี้เขาจะจัดห้องให้ลูกค้าหนึ่งคนพักหนึ่งห้อง เพื่อให้ลูกค้าพึงพอใจ

บรรทัดที่ 1 ประกอบด้วยจำนวนเต็ม T แทนจำนวนชุดทดสอบ โดยที่ $1 \leq T \leq 5$ ข้อมูลนำเข้า สำหรับแต่ละชุดทดสอบ

บรรทัดที่ 1 จำนวนเต็ม N แทนจำนวนลูกค้าที่ต้องการเข้าพัก โดยที่ $1 \leq N \leq 10^5$

บรรทัดที่ 2 จำนวนเต็ม N จำนวนคั่นด้วยช่องว่าง 1 ช่อง แทนวันที่ลูกค้ามาถึง

บรรทัดที่ 3 จำนวนเต็ม N จำนวนคั่นด้วยช่องว่าง 1 ช่อง แทนจำนวนวันที่ลูกค้าต้องการพัก

ข้อมลนำออก จำนวนห้องพักที่น้อยที่สุดที่เจ้าของโรงแรมต้องสร้าง

Input	Output
2	3
3	3
1 2 3	
3 3 3	
5	
1 2 3 4 5	
2 3 4 5 6	

G07: เศษส่วนอียิปต์ (Egyptian Fraction)

เศษส่วนที่เป็นบวกทุก ๆ จำนวนที่สามารถเขียนแทนในรูปของผลบวกของเศษส่วนโดยเศษมีค่าเท่ากับ 1 เท่านั้น (sum of unique unit fractions) เรียกเศษส่วนดังกล่าวว่า เศษส่วนอียิปต์ (Egyptian Fraction) ตัวอย่าง

เศษส่วนอียิปต์ 2/3 สามารถแทนด้วย 1/2 + 1/6

เศษส่วนอียิปต์ 6/14 สามารถแทบด้วย 1/3 + 1/11 + 1/231

เศษส่วนอียิปต์ 12/13 สามารถแทนด้วย 1/2 + 1/3 + 1/12 + 1/156

เราสามารถเขียนเศษส่วนอียิปต์โดยใช้ Greedy Algorithm ได้ดังนี้ สำหรับ 6/14

- หา ceiling ของ 14/6 ได้ 3 ดังนั้นเศษส่วนแรก คือ 1/3

- ทำซ้ำสำหรับ (6/14 – 1/3) จนได้ผลรวมทั้งหมดเท่ากับ 6/14 นั่นคือ ส่วน หาร เศษ เท่ากับ 0

a,b เป็นจำนวนเต็ม 2 จำนวน แทนเศษและส่วน คั่นด้วยช่องว่างหนึ่งช่อง ข้อมูลนำเข้า

แสดงเลขที่เป็นส่วนทั้งหมดของเศษส่วนอียิปต์ <u>ข้อมูลนำออก</u>

Input	Output
2 3	2
	6
6 14	3
	11
	231
12 13	2
	3
	12
	156



G09: Max Min

กำหนดให้ arr เป็นอาร์เรย์ของจำนวนเต็ม และ k เป็นจำนวนเต็ม โดยที่ subarr หรืออาร์เรย์ย่อยขนาด k สามารถสร้างจากสมาชิกของ arr ที่กำหนดให้

ให้นักเรียนหาผลต่างที่น้อยที่สุดของ subarr

คำนวณได้จาก max(subarr) - min(subarr)

โดยที่ max คือค่าที่มากที่สุดใน subarr

min คือค่าที่น้อยสุดใน subarr

ข้อมูลนำเข้า บรรทัดที่ 1 จำนวนเต็มสองจำนวน $m{n}$ และ $m{k}$ แทนจำนวนสมาชิกในอาร์เรย์และค่าเป้าหมาย

ตามลำดับ

บรรทัดที่ 2 ถึงบรรทัดที่ n+1 จำนวนเต็ม $\,n\,$ จำนวน

ข้อมูลนำออก แสดงผลต่างที่น้อยที่สุดของ subarr อาร์เรย์ขนาด k

ตัวอย่าง

) - 0

1 = 0	
k = 2	
1=1	
$\kappa = 3$	
1 = 2	
K = 4	

Input	Output	คำอธิบาย
7 B 10 B 100 Z5 300 Z0 200 100 1000 200 300 300 300 1000	20	เมื่อ $k=3$ อาร์เรย์ย่อยที่มีสมาชิก 3 ตัว ให้ผลต่างที่น้อย ที่สุดคือ 10, 20, 30 $\max(10,20,30)-\min(10,20,30)=20$
10 4 1 2 3 4 10 20 30 40 100 200	3	เมื่อ $k=4$ อาร์เรย์ย่อยที่มีสมาชิก 4 ตัว ให้ผลต่างที่น้อย ที่สุดคือ 1, 2, 3, 4 $\max(1,2,3,4)-\min(1,2,3,4)=3$
5 2 1 2 1 2 1	0	เมื่อ $k=2$ อาร์เรย์ย่อยที่มีสมาชิก 2 ตัว ให้ผลต่างที่น้อย ที่สุดคือ 1, 1 $\max(1,1) - \min(1,1) = 0$ $\max(2,2) - \min(2,2) = 0$